Automatic Classification of Points-of-Interest for Land-use Analysis

Filipe Rodrigues, Francisco C. Pereira, Ana Alves
Centre for Informatics and Systems of the University of Coimbra
University of Coimbra
Coimbra, Portugal
{fmpr,camara,ana}@dei.uc.pt

Shan Jiang, Joseph Ferreira
Department of Urban Studies and Planning
Massachusetts Institute of Technology
Boston, U.S.A.
{shanjang,jf}@mit.edu

***Abstract*—This paper describes a methodology for automatic classification of places according to the North American Industry Classification System. This taxonomy is applied in many areas, particularly in Urban Planning. The typical approach is to manually classify places/Points-of-Interest that are collected with field surveys. Given the financial costs of the task some semi-automatic approaches have been taken before, but they are still based on field surveys and official census. In this paper, we apply machine learning to fully automatize the classification of Points-of-Interest collected from online sources. We compare the adequacy of several algorithms to the task, using both flat and hierarchical approaches, and validate the results in the Urban Planning context.**

Keywords-machine learning; space analysis; points-of-interest; urban planning; GIS.

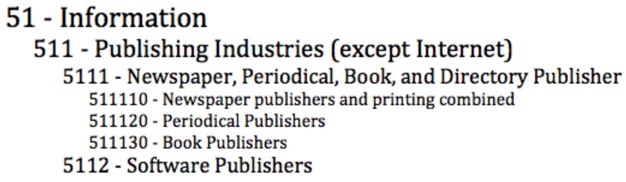
I. Introduction

A Point-of-Interest (or POI for short) is a specific point location that a considerable group of people find useful or interesting. POIs can be used in navigation systems, characterization of places, context-aware systems, city dynamics analysis, geo-referencing of texts, etc.

Despite its usefulness, the production of POIs is scattered across a myriad of different websites, systems and devices, thus making it extremely difficult to obtain an exhaustive database of such wealthy information. There are hundreds, if not thousands, of POI directories in the Web like Yahoo.com, Manta.com and YellowPages.com, each one using its own taxonomy of categories or tags. It is therefore essential to unify these different sources by mapping them to a common taxonomy, otherwise their application as a whole becomes impractical.

In this paper, we propose the use of machine learning techniques to automatically classify POIs from different sources to a standard taxonomy such as the North American Industry Classification System (NAICS) used in the U.S., Canada and Mexico, or the International Standard Industrial Classification (ISIC) used in the United Nations. Doing so is essential to allow a proper analysis of the POI data, especially when coming from different sources. A good example is the land-use analysis, which is a crucial task in Urban Planning. If the POIs do not share a common taxonomy we are not be able to determine, for instance, how many POIs of universities exist in a given area, since a POI source might classify them as "schools" and the other as "higher education". Although our approach would be similarly applicable to other classification standards, in this paper we are only interested in classifying POIs according to the North American Industry Classification System (NAICS).

The NAICS is the standard used by Federal statistical agencies in classifying business establishments for the purpose of collecting, analyzing, and publishing statistical data related to the U.S. business economy [1]. The NAICS was adopted in 1997 to replace the old Standard Industrial Classification (SIC) system. It is a two to six-digit hierarchical classification code system, offering five levels of detail. Each digit in the code is part of a series of progressively narrower categories, and more digits in the code signify greater classification detail. The first two digits designate the economic sector, the third digit designates the sub-sector, the fourth digit designates the industry group, the fifth digit designates the NAICS industry, and the sixth digit designates the national industry. A complete and valid NAICS code contains six digits [2]. Figure 1 shows part of the NAICS hierarchy.



51 - Information
- 511 - Publishing Industries (except Internet)
  - 5111 - Newspaper, Periodical, Book, and Directory Publisher
    - 511110 - Newspaper publishers and printing combined
    - 511120 - Periodical Publishers
    - 511130 - Book Publishers
  - 5112 - Software Publishers

Figure 1. Example of the NAICS hierarchy

After comparing several classification methods, we apply the results to the urban modeling task of estimating employment size at a disaggregated level. This task is traditionally made at a coarser level (Traffic Analysis Zone, Census Tract or Block Group level) than what could be now possible.

To the authors best knowledge, there is no previous work that automatically classifies POIs into the NAICS taxonomy. This is our main contribution.

The rest of this paper is organized as follows. Section II presents previous related studies. Section III explains

ISBN: 978-1-61208-178-6

our data analysis and modeling methodology, from data preparation to model generation and validation. Section IV shows the obtained results. In Section V we describe an application of this methodology to the field of Urban Planning. We finish the paper with some conclusions and further work.

II. State of the Art

The applications of machine learning algorithms in classification tasks are vast and cover diverse areas that range from Speech Recognition to Medicine, including forecasting in Economics and Environmental Engineering or Road Traffic Prediction. On the other hand, in Urban Planning, land-use/land-cover information has long been recognized as a very important material [3]. However, as Fresco [4] claimed, accurate data on actual land-use cannot be easily found at both global/continental and national/regional scales. In order to cope with these problems, automatic approaches to classify land-use are being developed using different techniques usually based on machine learning algorithms.

A common approach to infer land-use/land-cover is to use satellite imagery. However, while these approaches have already proven to get good results, they are more suited to land-cover inference, which is considered somehow different from land-use by many authors. Campbell [5], for example, considers land-cover to be concrete whereas land-use is abstract. That is, land-cover can be mapped directly from images, while land-use requires land-cover and additional information on how the land is used. Danoedoro [6] tries to improve land-use classification via satellite imagery by combining spectral classification, image segmentation and visual interpretation. Although he showed that satellite imagery could be used for generating socio-economic function of land-use at 83.63% accuracy, he is the first to recognize that applying such techniques to highly populated areas would be problematic.

Li et al. [7] use data mining techniques to discover knowledge from GIS databases and remote sensing image data that could be used for land-use classification. Using the C5.0 algorithm they get an accuracy of 89% in land-use classification.

An alternative to satellite imagery is the POI data. Using a large commercial POI database, Santos and Moreira [8] create and classify location contexts using decision trees. They identify clusters by means of a density-based clustering algorithm, which allow them to define areas (or regions) through the application of a concave hull algorithm they developed to the POIs within each cluster. Finally, making use of the C5.0 algorithm, they classify a given location according to such characteristics as the number of POIs in a cluster, the size of the area of the cluster and the categories of the POIs within the cluster.

In order to use POI data for the classification of places and land-use analysis, POI classification is an essential task. Griffin et al. [9] use decision trees to classify GPS-derived POIs. However, they refer to POIs as “personal” locations to a given individual (i.e., home, work, restaurant, etc.). The main goal of their approach is then to automatically classify trips. In their approach, they start by determining clusters of trip-stops (i.e., stops that took more than 5 minutes) using a density-based clustering algorithm (Dbscan). Then, they make use of the C4.5 algorithm to classify the generated clusters as being “home”, “work”, “restaurant”, etc., based on the time of the day and the length of the stay. However, no previous approaches have been made to classify POIs to a classification system such as NAICS. The latter is widely used for industry classification and has already been used, for instance, to classify Web Sites through machine learning techniques [10].

Spatial analysis has long been a topic of interest for researchers, who seek a comprehensive understanding on how the city behaves in different perspectives and its impact in the economy. Methods for analyzing spatial (and space-time) data have already been well developed by statisticians [11] and econometricians [12]. An interesting example is provided by Currid et al. [13], who try to understand the importance of agglomeration economies as a backbone to urban and regional growth, by identifying clusters of several “advanced” service sectors (professional, management, media, finance, art and culture, engineering and high technology) and comparing them in the top ten populous metropolitan areas in the U.S.

III. Approach

In this section we describe our approach, particularly what are the sources of our POI data, how we generate the training data, what methods we use for classification and how we perform validation.

A. POI Sources

Our data consists of a large set of POIs extracted from Yahoo! through their public API, another set acquired from Dun & Bradstreet (D&B) [14], a consultancy company that specializes in commercial information and insight for businesses, and a third one from InfoUSA.com provided by the Harvard Center for Geographic Analysis (ESRI Business Analyst Data). In the first data set (from Yahoo!), the database is essentially built from user contributions. In the other two the data acquisition process is semi-automatic and involves integration of official and corporate databases, statistical analysis and manual evaluation [14]. The POIs from D&B and InfoUSA have a NAICS code assigned (2007 version), which is not present in Yahoo!. However, each POI from Yahoo! is assigned, in average, roughly two arbitrary categories from the Yahoo! categories set. These categories are specified by the user, through a textfield and can be rather disparate since Yahoo! forces no restrictions over them, thus they can be seen as mere tags. Considering that every POI source provides either some categories or tags associated with their POIs, we take advantage of this information to classify them to the NAICS, where a single unifying code is assigned to each POI.

Our dataset contains 156364 POIs from Yahoo!, 29402 from D&B and 196612 from InfoUSA for the greater metropolitan area of Boston, Massachusetts. We also used 331118 POIs from Yahoo! and 16852 from D&B for the New York city area to see how our previously trained model would perform in a different city. We estimate that the Yahoo!'s categories taxonomy has more than 1300 distinct categories distributed along a 3-level hierarchy. On the other hand, NAICS has a total of 2332 distinct codes distributed along their 6-level hierarchy (1175 only in the sixth level).

Given its nature, the growth of the Yahoo! database (or any other user-content platform) is considerably faster than D&B and InfoUSA, and the POI categorization follows less strict guidelines, which in some cases, as mentioned before, may become subjective. This dynamic nature of these internet POI sources, together with the fact that they are publicly available to anyone and usually cover entire countries, make them extremely attractive. Our hypothesis is that there is considerable coherence between Yahoo! categories and NAICS codes, such that a model can be learned that automatically classifies incoming Yahoo! POIs.

B. POI Matching and Data Preparation

In order to generate training data for the machine learning algorithms we use a *POI Matching* algorithm, which compares POIs according to their name, Web Site and distance. It makes use of the JaroWinklerTFIDF algorithm [15] to identify close names, ignoring misspelling errors and some abbreviations. We set the similarity thresholds to high values in order to get only high confidence matches. By manually validating a random subset of the POI matches identified (6 sets of 50 random POIs assigned to 6 volunteers), we concluded that the percentage of correct similarities identified was above 98% ($\sigma = 1.79$). Differently to validations later mentioned in this paper, this is an extremely objective one, not demanding external participants or a very large sample[1].

After matching Yahoo! POIs to D&B and InfoUSA, we built two different geographic databases, where each POI contains a set of categories from Yahoo! and a NAICS classification provided by D&B and InfoUSA respectively. From this point on, we shall refer to the initial dataset, which results from POI matches between Yahoo! and D&B, as dataset A, and to the dataset resultant from the POI matching between Yahoo! and InfoUSA as dataset B. The later is six times larger than the former, due to larger coverage of InfoUSA in Boston.

Table I shows some statistic details of both datasets used.

Table I
Some statistics of datasets A and B

Dataset	A	B
NAICS source	D&B	InfoUSA
Total POIs	7289	44634
Distinct NAICS	504	689
Distinct Yahoo! categories	802	1109
Distinct Yahoo! category combinations	569	1002
Category combinations that appear only once	136	92
Categories that appear only once	181	107
NAICS that appear only once	115	96

The dataset A contains 7289 POIs for Boston and Cambridge and 2415 for New York. In comparison with the original databases, these are much smaller sets due to a very conservative POI matching approach (string similarity of at least 80%, max distance of 80 meters). However the POI quantities are high enough to build statistically valid models. We performed a detailed analysis of this data and identified 569 different category combinations, which included only 802 distinct categories from the full set (of over 1300). From D&B, our data covers 504 distinct six-digit NAICS codes. However, the 2007 NAICS taxonomy has a total of 1175 six-level categories, meaning that our sample data only covers some of the most common NAICS codes, which only represents about 43% of the total number of the NAICS categories. Nevertheless, the remaining ones are more exotic in our context and hence less significant for posterior analyses.

Further analyses on the coherence between NAICS and Yahoo! showed that only in 80,2% of the POIs in dataset A the correspondent NAICS was consistent with the most common one for that given set of categories, which means that about one fifth of the POIs are incoherent with the rest of the sample. This fact highlights the problem of allowing users to add arbitrary categories to their POIs without restrictions. For different NAICS levels, particularly for two-digit and four-digit NAICS, the same analyses showed, as expected, a higher level of coherency. For the two and four-digit NAICS, 87,1% and 83,4% of the POIs, respectively. Therefore, by having the same set of Yahoo! categories mapping to different NAICS codes in different occasions, it is not expectable that we obtain a perfect model that correctly classifies all test cases. In order to understand the impact of these inconsistencies in the results, we also modified the POI dataset so that the NAICS code of a given POI would match the NAICS codes of the other POIs with the same category set, assigning to each POI the most common NAICS code for that given category set in the dataset. The results of this separate experiment are also presented in Section IV.

Tables II and III show, respectively, the five most common NAICS and Yahoo! categories we identified in dataset A.

Regarding dataset B, we identified 689 distinct NAICS

[1] Using the central limit theorem, the standard error of the mean should be near 0.73. Assuming an underestimation bias for n=6 of 5% (by the [16]), accuracy keeps very high, yielding a 95% confidence interval of [96.5%, 98.7%]

Table II
MOST COMMON NAICS IN THE DATASET A

NAICS code	Description	Occurrences
423730	Warm Air Heating and Air-Conditioning Equipment and Supplies Merchant Wholesalers	707
446130	Optical Goods Stores	200
314999	All Other Miscellaneous Textile Product Mills	193
493120	Refrigerated Warehousing and Storage	136
332997	Industrial Pattern Manufacturing	123

Table III
MOST COMMON YAHOO! CATEGORIES IN THE DATASET A

Yahoo! category	Occurrences
Salons	157
All Law Firms	129
Government	116
Trade Organizations	115
Architecture	86

Table IV
BRIEF DESCRIPTION OF THE ALGORITHMS TESTED

Implementation	Description
ID3	Unpruned decision tree based on the ID3 algorithm.
C4.5	Pruned or unpruned C4.5 decision tree.
C4.5graft	Grafted C4.5 decision tree.
RandomForest	Forest of random trees, i.e., trees with K randomly chosen attributes at each node.
JRip	Propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), as proposed by W. Cohen as an optimized version of IREP.
IBk	K-nearest neighbors classifier that can do distance weighting.
IB1	1-nearest-neighbor classifier. Simplification of IBk.
K*	K* is an instance-based classifier. The class of a test instance is determined from the class of similar training instances . It uses an entropy-based distance function.
BayesNet	Bayesian Network
NaiveBayes	Naive Bayes model

codes and 1109 distinct categories of the more than 1300 that we found in Yahoo!. The latter are in larger number than the ones from dataset A (only 802) and therefore dataset B provides a better coverage of the source taxonomy. The number of distinct category combinations almost doubled when compared to dataset A, which leads to more diversity in the training data and probably to more accurate classifiers.

C. Flat Classification

The "flat classification" task corresponds to directly assigning a NAICS code to a POI given its "bag" of Yahoo! categories. It is "flat" because the inherent hierarchy of the NAICS is not taken into account in the classification model. Each NAICS code is simply seen as an isolated string "tag" that is assigned to a POI.

We experimented various machine learning algorithms for this particular classification task. Table IV provides a brief description of the algorithms we tested. It is beyond the scope of this paper to describe any of the algorithms in detail. The interested reader is redirected to dedicated literature (e.g., [17], [18]).

In our experiments we built classifiers for different NAICS levels (i.e., NAICS categories with different granularities), particularly two, four and six-digit NAICS codes. This choice is typical in Urban Planning depending on the study at hand (e.g., level 2 allows to analyze economic sectors, while level 6 goes to the level of the establishment specificities).

For validation purposes we use ten-fold cross-validation [17], [18]. We also performed validation with an external test set (data from a another city, New York) to understand the dependency of the generated models on the study area.

D. Hierarchical Classification

In this approach we take advantage of the hierarchical structure of the NAICS, thus the overall classifier is itself a hierarchy of classifiers. In this hierarchy each classifier decides what classifier to use next, narrowing down the NAICS code possibilities on each step, until a final 6-digit code (or 4-digit code, depending on the goal) is achieved. Figure 2 depicts one possible hierarchy.

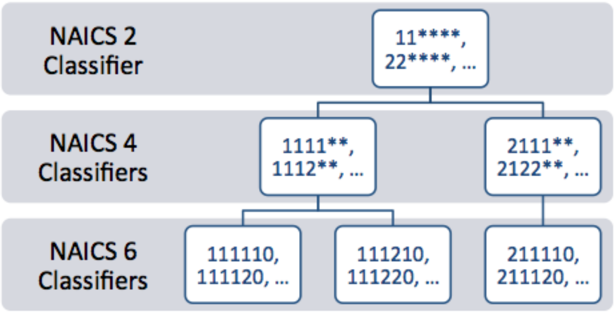


Figure 2. A possible hierarchy of classifiers

By looking at the hierarchy above, we can see that it has 3 levels (2, 4 and 6-digit NAICS). The first level always consists of a single classifier that decides which NAICS economic sector (2-digit code) the POI belongs to. Taking the sector into account, the algorithm then decides which classifier to use next at the second level. After that, the same process repeats until a leaf node is achieved in the tree structure of the hierarchy of classifiers. To provide an example consider a POI that has the following NAICS code: 111110. According to Figure 2 the top-level classifier will decide that it belongs to sector 11 ("Agriculture, Forestry, Fishing and Hunting") and the left-most level 2 classifier will be used next. Then, this classifier will determine that the 4-digit NAICS code of the POI is 1111 ("Oilseed and Grain Farming") and, based on this decision, the left-most classifier in the third level of the figure will be used, and will supposedly classify the POI with the NAICS code 111110 ("Soybean Farming"). Of course along this top-down process a mistake can be made by one classifier. In this case, the error would propagate downwards and there would be no way to recover from it, and hence the final NAICS code would be wrong.

Our hypotheses is that by using a hierarchy of classifiers, the classification task will be divided into several classification models, each one less complex, more accurate and dealing with a simpler problem. If we consider, for example, the ID3 algorithm, the entropy values for the different features will be computed according to a smaller class subset, and therefore the selection of the next feature to use (which is based on the entropy calculation) will be different and the resulting tree will also be different. Hopefully, the generated classifier will be more suited to that particular classification (like deciding for a POI if it belongs to the subcategory 531, 532, etc, knowing that it belongs to the NAICS sector 53).

In our experiments we use three different hierarchies of classifiers, two with 2 levels:

- NAICS 2 and NAICS 4
- NAICS 2 and NAICS 6

and other one with 3 levels:

- NAICS 2, NAICS 4 and NAICS 6

As we did for the flat classification, we also tried to test different types of machine learning algorithms: bayesian networks, tree-based learners, instance-based learners and rule-based learners. Neural networks were not possible to test due to their computational demands, both in processing power and memory.

For the hierarchical approaches we also perform ten-fold cross-validation, but the data splitting between training/testing is more prone to biased results than with standard flat classification. As in normal ten-fold cross-validation, we also start by leaving 10% of the data out for test and use the remaining 90% for training, repeating this process ten times. However, each classifier in a given level only receives the part of those 90% of training data that respects to it. For instance, a level two classifier for deciding which subcategory of the NAICS sector 53 a given POI belongs to would only be trained with POIs that belong to that NAICS sector. Hence, the only classifier that receives all the training data (90%) would be the top-level classifier (i.e., the one that decides which NAICS sector a POI belong to). After the training phase, the hierarchy is tested with the 10% of the data left out. This process is repeated ten times, and the average accuracy over the ten iterations is determined.

IV. Results

Table V shows the accuracies obtained using different machine learning algorithms in a "flat" setting for different NAICS levels (two, four and six-digit codes) for dataset A. We can see that the tree-based (e.g., ID3, RandomForest) and instance-based learning approaches (e.g., IBk, K*) are the ones that perform better in this classification task, especially the latter. Notice that at the sixth-level only 80,2% of the NAICS codes in the data were assigned in a totally non-ambiguous way. The most successful algorithm is IBk (with k=1), which essentially finds the similar test case and assigns the same NAICS code. The difference in accuracy between tree-based and instance based approaches is too small to conclude which one outperforms the other, however we could expect that instance based models bring better results since the distribution of the different Yahoo! categories is relatively even among examples of the same NAICS code (implying no clear "dominance" of some categories over others). Understandably, the Naive Bayes algorithm performs badly because the assumption that different Yahoo! categories for the same NAICS classification are independently distributed is obviously false (e.g., "Doctors & Clinics, Laboratories, Medical Laboratories" are correlated). Such assumption is not fully necessary in Bayesian Networks, which actually brings better results. Unfortunately, we could not find a model search algorithm that performs in acceptable time (less than 72 hours) and produces a more accurate model. We used Simulated Annealing and Hill Climbing.

Table V
Accuracies obtained by different machine learning algorithms with POIs from dataset A for the Boston area

Algorithm	NAICS2(kappa)	NAICS4(kappa)	NAICS6(kappa)
ID3	85.495 (0.842)	77.955 (0.776)	74.015 (0.737)
C4.5	84.241 (0.828)	77.630 (0.772)	73.071 (0.727)
Random Forest	86.174 (0.849)	79.298 (0.789)	74.753 (0.744)
JRip	81.334 (0.795)	74.340 (0.737)	69.264 (0.686)
IB1	82.736 (0.812)	74.266 (0.738)	68.644 (0.683)
IBk	86.646 (0.854)	79.475 (0.791)	75.343 (0.750)
K*	85.702 (0.844)	79.726 (0.794)	75.387 (0.751)
BayesNet	80.950 (0.790)	56.721 (0.554)	45.064 (0.438)
NaiveBayes	74.399 (0.715)	40.446 (0.382)	30.264 (0.283)

As expected, we obtained better results classifying POIs with two-level NAICS codes than with the six-level NAICS codes, since the noise due to ambiguous NAICS codes assignments in the POI dataset is smaller (we now have 87.1% of non-ambiguous cases; see Section III-B).

In Table VI we can see the results obtained by changing the POI dataset A so that the NAICS codes of POIs where ambiguities arise are grouped together in the same "super-category", eliminating the inconsistencies.

Table VI
Accuracies obtained by different machine learning algorithms using a re-classified version of dataset A

Algorithm	NAICS2	NAICS4	NAICS6
ID3	92.975	89.728	88.680
RandomForest	93.609	90.805	89.846
IBk	94.170	91.189	89.979

By comparing the results in Table VI with the results in Table V, we realize that the NAICS labeling inconsistencies in the POI data have a major negative effect in the performance of the machine learning algorithms, reducing the accuracy in more than 16% in some cases for the six-level NAICS codes. This also gives indications for future versions of the NAICS, where some categories may become aggregated according to these “super-categories”.

It would be expectable to obtain accuracies closer to 100% for the results in Table VI. However, that does not happen since 115 of the 514 NAICS codes covered by our dataset A only occur once. Therefore, when we split the dataset to perform the ten-fold cross-validation, a significative number of the test cases will have NAICS codes that were not observed during training, causing the algorithm to incorrectly classify them.

Table VII shows the results we obtained by training the machine learning approaches with dataset A from Boston and Cambridge and testing them with New York POI data. As we can see in the results, if we apply the generated model to a different city, it still performs well, even though the accuracy drops a small amount in some cases. This is understandable since even the Yahoo! taxonomy differs slightly from city to city.

Table VII
ACCURACIES OBTAINED BY DIFFERENT MACHINE LEARNING ALGORITHMS USING POI DATA FROM BOSTON FOR TRAINING AND POI DATA FROM NEW YORK FOR TESTING

Algorithm	NAICS2	NAICS4	NAICS6
ID3	85.061	75.586	70.209
RandomForest	85.488	76.867	71.318
IBk	85.360	76.909	71.276

Table VIII shows the results obtained for the different machine learning algorithms using dataset B. By analyzing these results we can see that the classification accuracies have significantly improved over dataset A, which shows the importance of the training data size in the performance of the machine learning algorithms.

Table VIII
ACCURACIES OBTAINED BY DIFFERENT MACHINE LEARNING ALGORITHMS WITH POIS FROM DATASET B FOR THE BOSTON AREA

Algorithm	NAICS2(kappa)	NAICS4(kappa)	NAICS6(kappa)
ID3	90.567 (0.897)	85.459 (0.852)	82.091 (0.819)
C4.5	90.113 (0.800)	85.085 (0.849)	81.831 (0.816)
RandomForest	90.758 (0.899)	85.710 (0.855)	82.436 (0.823)
JRip	85.748 (0.844)	80.998 (0.807)	78.495 (0.780)
IB1	87.224 (0.861)	81.495 (0.812)	76.826 (0.766)
IBk	91.024 (0.902)	85.974 (0.858)	82.553 (0.824)
K*	90.227 (0.893)	85.849 (0.856)	82.522 (0.824)
BayesNet	88.961 (0.880)	77.964 (0.776)	67.877 (0.675)
NaiveBayes	87.910 (0.868)	70.250 (0.696)	56.052 (0.554)

Finally Tables IX to XI show the results obtained using the different hierarchical classification schemes for various types of machine learning algorithms. There are some missing results in the cases where the algorithm took over 72 hours to run.

Table IX
COMPARISON BETWEEN THE RESULTS FOR DATASET B USING FLAT CLASSIFICATION (4-DIGIT NAICS) AND HIERARCHICAL CLASSIFICATION WITH 2 LEVELS (NAICS 2 AND 4)

Algorithm	Flat classification accuracy	Hierarchical classification Level1 acc.	Level2 acc.
ID3	85.459	90.659	85.620
C4.5	85.085	90.172	84.901
RandomForest	85.710	90.959	85.969
JRip	80.998	85.806	80.440
IB1	81.495	87.637	81.126
IBk	85.974	91.080	86.097
K*	85.849	90.305	85.244
BayesNet	77.964	88.002	74.243
NaiveBayes	70.250	30.688	20.091

Table X
COMPARISON BETWEEN THE RESULTS FOR DATASET B USING FLAT CLASSIFICATION (6-DIGIT NAICS) AND HIERARCHICAL CLASSIFICATION WITH 2 LEVELS (NAICS 2 AND 6)

Algorithm	Flat classification accuracy	Hierarchical classification Level1 acc.	Level2 acc.
ID3	82.091	90.659	82.100
C4.5	81.831	90.173	81.484
RandomForest	82.436	90.959	82.477
JRip	78.495	85.806	76.398
IB1	76.826	87.637	76.826
IBk	82.553	91.080	82.551
K*	82.522	90.305	81.661
BayesNet	67.877	89.059	69.336
NaiveBayes	56.052	88.002	59.885

Table XI
COMPARISON BETWEEN THE RESULTS FOR DATASET B USING FLAT CLASSIFICATION (6-DIGIT NAICS) AND HIERARCHICAL CLASSIFICATION WITH 3 LEVELS (NAICS 2, 4 AND 6)

Algorithm	Flat classification accuracy	Hierarchical classification Level1 acc.	Level2 acc.	Level3 acc.
ID3	82.091	90.659	85.620	82.111
C4.5	81.831	90.172	84.901	81.341
Random Forest	82.436	90.959	85.969	82.398
JRip	78.495	85.806	80.440	76.889
IB1	76.826	87.637	81.126	76.826
IBk	82.553	91.080	86.097	82.539
K*	82.522	90.305	85.244	81.486
BayesNet	67.877	-	-	-
NaiveBayes	56.052	-	-	-

Our intuition was that hierarchical classification would perform generally better than standard flat classification. However, only in some algorithms the results improved. Therefore, we will not argue that hierarchical classification of POIs into the NAICS is always a better solution. In fact, as shown before by comparing the datasets A and B, the quality and the dimensions of the dataset seems to have a much bigger impact on the results than whether we apply hierarchical or flat classification.

Another interesting fact in the results from the hierarchical classification is that the accuracies vary considerably with the hierarchy type used. For instance, when classifying POIs with 6-digit NAICS codes, we can see that using a two-level hierarchy the RandomForest algorithm improved over the flat classification, while using a three-level hierarchy it became worse (although the differences in accuracy are small). One of the possible cause for this, is that the hierarchy type used directly affects the number of training instances at each node of the hierarchy tree, and depending on the machine learning algorithm, the number of training instances will have different impacts on the results.

V. An application in Urban Planning

In this section we describe a practical application of Yahoo! POIs classified to the NAICS using a non-hierarchical approach with the k-nearest neighbor classifier (see Section III-C for more details).

In the field of Urban Planning, urban simulation models have evolved significantly in the past several decades. For instance, the travel demand modeling approach has been evolveing from the traditional Four-Step Model (FSM) to the Activity-Based Model (ABM) [19]. Consequently, requirements for disaggregated data increase greatly, ranging from population data, employment data, to travel survey data. The employment data (on the travel destination side) is usually obtained from proprietary sources, which adds another layer of barriers to widely applying the Activity-Based Modeling approach, let alone the expensive travel-survey data acquisition. In order to study this issue, researchers are trying to develop new methods of estimating disaggregated employment size and location by category.

In our case, we intend to develop a set of new methods and demonstrate their applications for estimating activities, incorporating them into travel demand and urban simulation models. This will be beneficial for cities that lack detailed survey data for building Activity-Based Models but wish to test the sensitivity of travel behavior to policy changes such as Intelligent Transportation Systems (ITS) implementations that are likely to alter activity patterns. An important step to achieve these goals is to obtain a disaggregated employment distribution by POIs of an area. For the case of Cambridge, MA, we have official data at the Block Group (BG) level (obtained from the U.S. Census Transportation Planning Package 2000), which essentially describes the total size of employees by economic sector at that spatial resolution. We need to distribute these totals into Block or Parcel level.

For demonstration purposes we only use POIs from the "Retail Trade" sector of the NAICS taxonomy, i.e., categories whose code starts by 44 or 45. Figures 3 and 4 show the aggregated retail employment density at the Block Group level and distribution of our POI data from Yahoo! at the Census Block level for Cambridge, respectively.

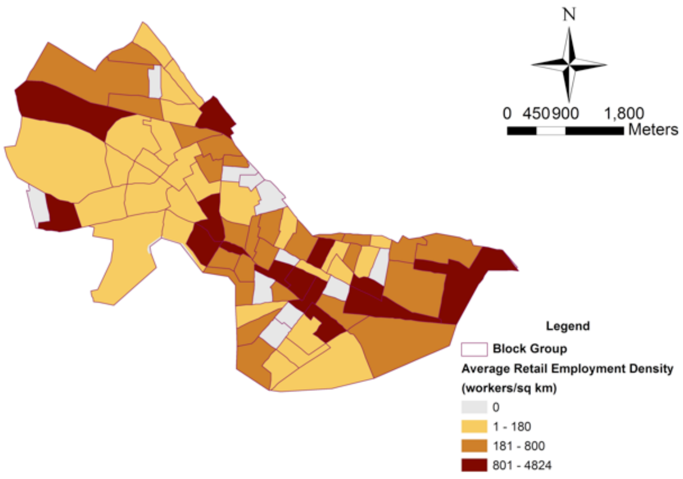


Figure 3. Aggregated retail employment density at the Block Group level (pl/sq km= employed people per square kilometer).

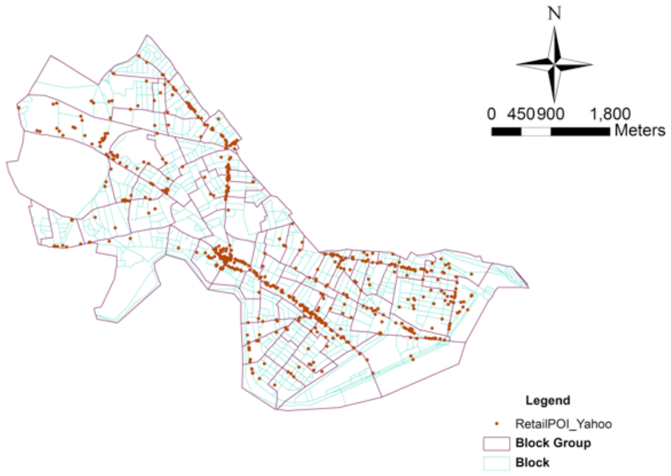


Figure 4. Cambridge retail POI distributions from Yahoo!

By using the business establishment survey data (from InfoUSA, 2007), which is believed to be close to the population, we are able to obtain a benchmark estimate of employment size by category at the Census Block level for the study areas. This will function as a ground truth to test our algorithm. Notice however that the dates for each of the databases are quite distinct (2000 for Census, 2007 for InfoUSA and 2010 for Yahoo!) therefore some error is expected to happen.

We employ the local maximum likelihood estimation (MLE) method as described below to derive the disaggregated destination estimation at Block level.

1) We calculate the total number of POIs (destinations) by category c in each Block b.

2) We assume that the employment size at destination d in Block Group g of category c is proportional to some function f of its associated block area $a_{d,c,g}$, which means the effective area of the destination d in Block Group g of category c. The form of function f will be explored based on the empirical data, and we also allow the possibility that $f(a_{d,c,g}) = a_{d,c,g}$, which is the natural benchmark case. Mathematically, assume that for employment category c, there are $n_{c,g}$ destinations at Block Group g. For $d = 1, 2, \ldots, n_{c,g}$, let the random variable $e_{d,c,g}$ be the employment size of category c at destination d in Block Group g.
3) We assume that $e_{d,c,g}(d = 1, 2, \ldots, n_{c,g})$ are i.i.d.$(f(a_{d,c,g}) \cdot \alpha_{c,g}, \sigma^2_{c,g})$, where $\alpha_{c,g}$ is the employment size of category c per unit of effective area at Block Group g; $\alpha_{c,g}$ and $\sigma_{c,g}$ are positive constants independent of d. $E(e_{d,c,g}) = f(a_{d,c,g}) \cdot \alpha_{c,g}$ and $Var(e_{d,c,g}) = \sigma^2_{c,g}$. We then estimate $\alpha_{c,g}$ by employing the maximum likelihood method locally at Block Group g for employment category c. Thus we obtain an estimate of employment size $e_{d,c,g}$ of category c at destination d in Block Group g.
4) Finally, we sum up the employment size in category c in Census Block b in Census Block Group g.

By employing the same local maximum likelihood method described above and using the business establishment survey data (e.g., ESRI Business Analysis package), which is believed to be close to the population POIs, we obtain a benchmark estimate of employment size by category at the Block level for the study area, $E^*_{b,c,g}$. By using the derived POI information (obtained from the machine learning algorithm), we obtain an estimate of employment size by category c at Block b for the study area, $\hat{E}_{b,c,g}$.

Then the mean squared error (MSE), weighted mean squared error (WMSE), and the relative weighted mean squared error (RWMSE) can be calculated to evaluate the goodness of fit of the model (see Equations 1, 2, 3, and 4).

$$MSE(\hat{E}_{b,c,g}, E^*_{b,c,g}) = \sum_{b,c,g} (\hat{E}_{b,c,g} - E^*_{b,c,g})^2 \tag{1}$$

$$WMSE(\hat{E}_{b,c,g}, E^*_{b,c,g}) = \sum_{b,c,g} w_{b,c,g} (\hat{E}_{b,c,g} - E^*_{b,c,g})^2 \tag{2}$$

$$RWMSE(\hat{E}_{b,c,g}, E^*_{b,c,g}) = \frac{\sum_{b,c,g} w_{b,c,g} (\hat{E}_{b,c,g} - E^*_{b,c,g})^2}{\sum_{b,c,g} w_{b,c,g} (\bar{E}_{b,c,g} - E^*_{b,c,g})^2} \tag{3}$$

$$\bar{E}_{b,c,g} = \frac{w'_{b,g} \sum_q E^*_{q,c,g}}{\sum_q w'_{q,g}} \tag{4}$$

Weights $\{w_{b,c,g}\}$ are normalized to reflect the proportion of each Census Block in the whole map. In Equation 2, when we take the weight $w_{b,c,g} = 1$ for any subscripts b, c, and g, the corresponding WMSE becomes MSE. In Equation 4, $w'_{b,g}$ = area of Block b in Block Group g, and $\bar{E}_{b,c,g}$ is the estimated employment size in Block b of category c, using the traditional disaggregation approach, assuming that the employment is uniformly distributed across blocks in each Block Group g.

If RWMSE is less than 1, it means that the quality of the derived POIs is reliable, so is the new method; the smaller the RWMSE, the more accurate is the method. If WMSE or RWMSE equals to 0, it means that the derived POIs from the Internet match exactly with the trusted proprietary POIs (treated as the population POIs). However, if RWMSE is greater than 1, it means that the derived POIs cannot well reflect the distribution of the population POIs.

Figures 5 and 6 show the estimation results of the disaggregated retail employment density at Block level in Cambridge, MA, by using POIs from infoUSA and Yahoo! respectively. By comparing the estimation results, we find that the disaggregated employment estimations by using the POIs captured from the Internet using Yahoo! and those obtained from the proprietary source (infoUSA 2007) are very close.

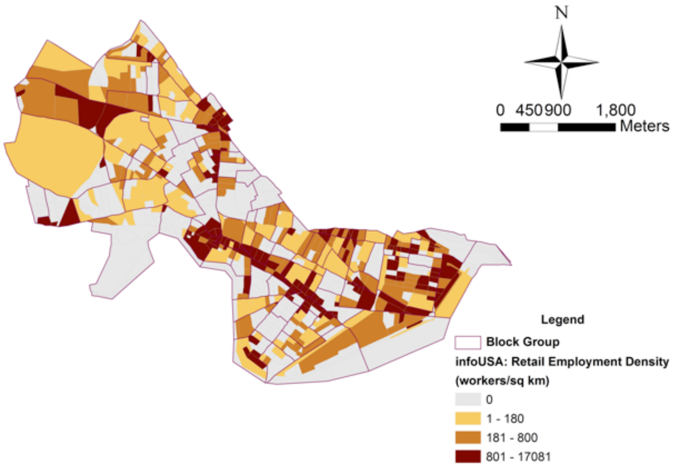


Figure 5. Disaggregated retail employment densities at the Block level, in Cambridge, MA, by using POIs from infoUSA

Employing Equation 3, the disaggregated employment estimation at the Block level using Yahoo! POI gives RMSE = 0.312. The RMSE is significantly smaller than 1, which means that using the extracted Yahoo! online POIs to estimate the disaggregated employment sizes at the Block level has reduced the mean squared error by around 69% compared to the traditional average disaggregation approach.

VI. Conclusion

In this paper, we showed that it possible to classify POI to the widely used NAICS system with several different machine learning algorithms using only the categories or tags that are commonly associated with them. We matched two different POI databases (InfoUSA and Dun & Bradstreet) to Yahoo!, in order to build two reliable training sets that have POIs with user provided bags of categories

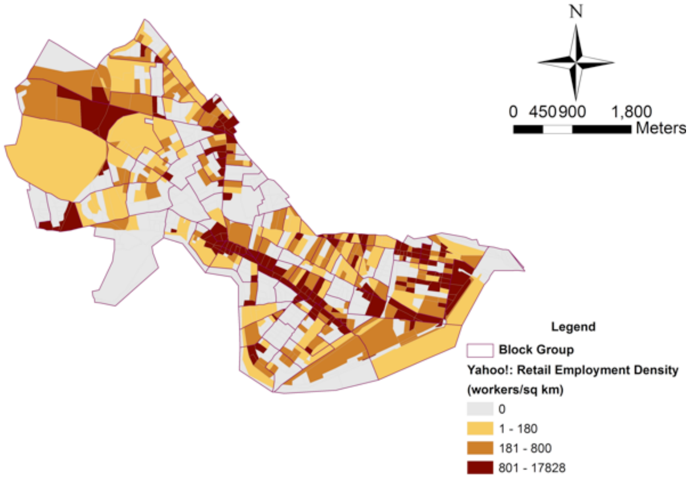


Figure 6. Disaggregated retail employment densities at the Block level, in Cambridge, MA, by using POIs from Yahoo!

classified with NAICS codes. We tested several classification algorithms and the results show that the best approaches for this particular task are inductive based algorithms, namely instance based and tree based learning. These allow for an accuracy as high as 82% in the most complex task (classification with 6-digit NAICS codes). We also tried to perform classification in a hierarchical way, however the results did not showed many improvements over the flat approaches, leading us to the conclusion that the size of the training set and its consistency/quality can have a larger impact on the results than the classification algorithm itself (except maybe for Bayesian approaches).

The classified POIs were applied to the urban modeling task of employment size and location disaggregation from Block Group level to Block level and the results show encouraging quality. This strengthens the idea that well classified POI data to a convenient taxonomy like the NAICS is of great use and can have many distinct applications.

To the authors best knowledge, this is the only work that proposes an automatic approach for classifying POIs to the NAICS, and therefore a comparison with other works is not possible. Thus, we contribute with a novel approach to this important problem that has high impact in urban modeling and space classification.

References

[1] U. C. Bureau. North american industry classification system (naics): Introduction, <Retrieved: November 2011>. http://www.census.gov/eos/www/naics/.

[2] N. Association. NAICS association: FAQ, <Retrieved: November 2011>. http://www.naics.com/faq.htm.

[3] D. T. Lindgren. *Land-use Planning and Remote Sensing*. Martinus-Nijhoff, Boston, MA, 1985.

[4] L. O. Fresco. *The Future of the Land – Mobilizing and Integrating Knowledge for Land-use Options*. John Wiley & Sons, Chichester, 1997.

[5] J. B. Campbell. *Mapping the Land – Aerial Imagery for Land use Information*. Association of American Geographers, Washington, D.C., 1983.

[6] P. Danoedoro. Extracting land-use information related to socio-economic function from quickbird imagery: A case study of semarang area, indonesia. *Map Asia 2006*, 2006.

[7] D. Li, K. Di, and D. Li. Land use classification of remote sensing image with GIS data based on spatial data mining techniques. *Geo-Spatial Information Science*, pp. 30-35, 2000.

[8] M. Santos and A. Moreira. Automatic classification of location contexts with decision trees. *CSMU-2006 : Proceedings of the Conference on Mobile and Ubiquitous Systems, Guimares, Portugal*, pp. 79-88, 2006.

[9] T. Griffin, T. Huang, and R. Halverson. Computerized trip classification of GPS data. *Proceedings of 3rd International Conference on Cybernetics and Information Technologies, Systems and Applications (CITSA 2006)*, pp. 22-30, 2006.

[10] J. Pierre. On the automated classification of web sites. *Linkoping Electronic Articles in Computer and Information Science*, 6, pp. 1-12, 2001.

[11] R. P. Haining. *Spatial data analysis in the social and environmental sciences*. Cambridge University Press, Cambridge, 1990.

[12] L. Anselin and R. Florax. *New directions in spatial econometrics*. Springer, New York, 1995.

[13] E. Currid and J. Connolly. Patterns of knowledge: The geography of advanced services and the case of art and culture. *Annals of the Association of American Geographers*, pp. 414-434, 2008.

[14] D. . Bradstreet. D & B Website, <Retrieved: November 2011>. http://www.dnb.com/.

[15] W. Cohen, P. Ravikumar, and S. Fienberg. A comparison of string distance metrics for name-matching tasks. *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-03), Acapulco, Mexico*, pp. 73-78, 2003.

[16] J. Gurland and R. Tripathi. A simple approximation for unbiased estimation of the standard deviation. *American Statistician*, pp. 30-32, 1971.

[17] T. M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[18] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[19] M. McNally and C. Rindt. *The Activity-Based Approach*. Handbook of Transportation Modeling. Elsevier, Amsterdam, London, 2008.